

## IN THE CLAIMS

This listing of claims replaces all prior listings:

1. (previously presented) A method in a data processing system for synchronizing calls at a client in a server and client system, comprising the steps of:

receiving from the server a plurality of service calls generated by a plurality of threads executed at the server;

receiving a synchronization call from the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change; and

placing at least one of said service calls associated with said synchronization call into a wait position, when said number of service calls indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ.

2. (original) The method according to claim 1, wherein said service calls are associated with said synchronization call by one of including respective identifiers into said at least one of said synchronization call and said service calls, and indicating one of a specific reception sequence and order of service of said service calls and said at least one synchronization call at the client.

3. (original) The method of synchronizing calls according to claim 1, wherein said receiving steps include receiving a first call sequence of a plurality of call sequences from the server, said first call sequence including a first synchronization call and at least one service call from a first thread, said first synchronization call including a first server call counter value indicating a first number of service calls executed at the server prior to the first synchronization call;

said method further comprising the step of:

comparing said first server call counter value with a client call counter value, said client call counter value indicating a second number of service calls executed at the client prior to receiving said first synchronization call; and

one of:

executing said first number of service calls of said first call sequence and counting said executed first number of service calls using a client call counter value, if said client call counter value and said first server call counter value coincide; and

placing said first call sequence into a wait position, if said client call counter value and said first server current call counter value differ.

4. (original) The method according to claim 1, wherein said service calls are generated asynchronously.

5. (original) The method according to claim 3, further comprising the steps of:

determining whether a second call sequence in a wait position is available, said second call sequence including a plurality of service calls from a second thread executed at the server and a second synchronization call including a second server call counter value indicating a third number of service calls executed at the server prior to said second synchronization call;

wherein if said second call sequence in a wait position is not available, waiting to receive further service calls and synchronization calls; and

wherein if said second call sequence is available, determining that said second server call counter value coincides with said client call counter value, and executing said third number of service calls of said second call sequence and incrementing said client counter value for each executed third number of service calls.

6. (original) The method of according to claim 5, further comprising the step of:

waiting for a third call sequence to be received from the server unit, the third call sequence including a third synchronization call including a third server call counter value coinciding with said client call counter value.

7. (original) The method according to claim 3, wherein said call sequences are received as groups included into packets from the server, each group being generated upon one of a timer signal at the server, a synchronous call at the server, and a synchronization call at the server.

8. (original) The method according to claim 2, wherein said synchronization call and said service calls are received in an arbitrary order.

9. (original) The method according to claim 1, wherein said service calls from said plurality of threads at the server are executed in corresponding threads at the client.

10. (original) The method according to claim 3, wherein said first server call counter value indicates a total number of service calls at the server executed prior to a current service call and requires communication with the client; and

wherein said client call counter value indicates a total number of service calls executed at the client and involves communication with the server.

11. (original) The method according to claim 1, wherein each of said service calls from the server includes at least one of:

obtaining instructions to display information on a display of the client;

rendering instructions;

storing instructions to store information at the client; and

information on processing results from the server.

12. (original) A method in a data processing system for synchronizing calls at a server in a server and client system, comprising the steps of:

transmitting a plurality of service calls generated by a plurality of threads at the server to a client;

generating a synchronization call when a thread of said plurality of threads executed at the server changes, said synchronization call indicating a number of service calls generated by said plurality of threads at the server prior to the thread change; and

transmitting said synchronization call to the client to allow the client to synchronize a service call execution.

13. (original) The method according to claim 12, wherein said service calls are associated with said synchronization call by one of including respective identifiers into said at least one of said synchronization call and said service calls, and indicating one of a specific reception sequence and order of service of said service calls and said at least one synchronization call at the client.

14. (original) The method according to claim 12, wherein said service calls are generated asynchronously.

15. (original) The method according to claim 12, further comprising the steps of:  
generating a current service call by a first thread executed at the server;  
determining a first thread identifier of a first thread and comparing said first thread ID with a second thread identifier of a second thread which issued a service call preceding said current service call;

wherein, if said first thread identifier and said second thread identifier differ, generating a first synchronization call including a server call counter value indicating a number of service calls executed at the server prior to said current service call and transmitting said first synchronization call to the client, for enabling the client to synchronize an execution of a plurality of service calls from at least said first thread and said second thread; and

counting said current service call using said server call counter value if said first thread identifier and said second thread identifier do not differ.

16. (original) The method according to claim 15, wherein a plurality of service calls from said first thread and said synchronization call comprise a call sequence; and

wherein said call sequences are received as groups included into packets from the server, each group being generated upon one of a timer signal at the server, a synchronous call at the server, and a synchronization call at the server

17. (original) The method according to claim 15, wherein said synchronization call includes said second thread identifier of said second thread, and said number of service calls include a thread identifier of each thread generating said service call; and

wherein said synchronization call and said number of service calls are transmitted to the client in an arbitrary order.

18. (original) The method according to claim 15, wherein said service calls from said plurality of threads at the server are executed in corresponding threads at the client.

19. (original) The method according to claim 15, wherein said server call counter value indicates a total number of service calls requiring communication with the client executed at the server, prior to the current service call.

20. (original) The method according to claim 12, wherein each service call from the server includes at least one of:

- obtaining instructions to display information on a display of the client;
- rendering instructions;
- storing instructions to store information at the client; and
- information on processing results from the server.

21. (original) The method according to claim 12, wherein a synchronization call is further generated upon an occurrence of one of the group comprising:

- a timer signal;
- a predetermined number of service calls; and
- a synchronous call.

22. (original) A method in a data processing system for synchronizing calls in a client and server system, the method comprising the steps of:

- transmitting a plurality of service calls generated by a plurality of threads executed at the server to the client;

- generating a synchronization call at the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change;

- transmitting said synchronization call to the client to allow the client to synchronize a service call execution;

- receiving said synchronization call at the client; and

- placing at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ.

23. (original) A computer readable medium containing instructions that cause a data processing system to perform a method of synchronizing calls in a client and a server system, the method comprising the steps of:

transmitting a plurality of service calls generated by a plurality of threads executed at the server to the client;

generating a synchronization call at the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change;

transmitting said synchronization call to the client to allow the client to synchronize a service call execution;

receiving said synchronization call at the client; and

placing at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ.

24. (previously presented) The computer readable medium according to claim 23, wherein said service calls are associated with said synchronization call by one of including respective identifiers into said at least one of said synchronization call and said service calls, and indicating one of a specific reception sequence and order of service of said service calls and said at least one synchronization call at the client.

25. (previously presented) The computer readable medium according to claim 24, wherein said receiving step includes receiving a first call sequence of a plurality of call sequences from the server, said first call sequence including a first synchronization call and at least one service call from a first thread, said first synchronization call including a first server call counter value indicating a first number of service calls executed at the server prior to the first synchronization call;

said method further comprising the step of:

comparing said first server call counter value with a client call counter value, said client call counter value indicating a second number of service calls executed at the client prior to receiving said first synchronization call; and

one of:

executing said first number of service calls of said first call sequence and counting said executed first number of service calls using a client call counter value, if said client call counter value and said first server call counter value coincide; and

placing said first call sequence into a wait position, if said client call counter value and said first server current call counter value differ.

26. (previously presented) The computer readable medium according to claim 24, wherein said service calls are generated asynchronously.

27. (previously presented) The computer readable medium according to claim 26, further comprising the steps of:

determining whether a second call sequence in a wait position is available, said second call sequence including a plurality of service calls from a second thread executed at the server and a second synchronization call including a second server call counter value indicating a third number of service calls executed at the server prior to said second synchronization call;

wherein if said second call sequence in a wait position is not available, waiting to receive further service calls and synchronization calls; and

wherein if said second call sequence is available, determining that said second server call counter value coincides with said client call counter value, and executing said third number of service calls of said second call sequence and incrementing said client counter value for each executed third number of service calls.

28. (previously presented) The computer readable medium according to claim 27, further comprising the step of:

waiting for a third call sequence to be received from the server unit, the third call sequence including a third synchronization call including a third server call counter value coinciding with said client call counter value.

29. (previously presented) The computer readable medium according to claim 26, wherein said call sequences are received as groups included into packets from the server, each group being generated upon one of a timer signal at the server, a synchronous call at the server, and a synchronization call at the server.

30. (previously presented) The computer readable medium according to claim 26, wherein said synchronization call and said service calls are received in an arbitrary order.

31. (previously presented) The computer readable medium according to claim 24, wherein said service calls from said plurality of threads at the server are executed in corresponding threads at the client.

32. (previously presented) The computer readable medium according to claim 26, wherein said first server call counter value indicates a total number of service calls at the server executed prior to a current service call and requires communication with the client; and

wherein said client call counter value indicates a total number of service calls executed at the client and involves communication with the server.

33. (previously presented) The computer readable medium according to claim 24, wherein each of said service calls from the server includes at least one of:

- obtaining instructions to display information on a display of the client;
- rendering instructions;
- storing instructions to store information at the client; and
- information on processing results from the server.

34. (original) A data processing system for synchronizing calls in a client and server system, the data processing system comprising:

- a client computer comprising:

- a memory including a client program that receives a plurality of service calls generated by a plurality of threads executed at the server, that receives a synchronization call from the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change, and that places at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ; and



a processor that runs said client program;

a server computer comprising:

a memory including a server program that transmits a plurality of service calls generated by a plurality of threads at the server to the client, that generates a synchronization call when a thread of said plurality of threads executed at the server changes, said synchronization call indicating a number of service calls generated by said plurality of threads at the server prior to the thread change, and that transmits said synchronization call to the client to allow the client to synchronize a service call execution; and

a processor that runs said server program; and

a network connecting said client computer and said server computer.

35. (original) An apparatus for synchronizing calls in a client and server system, the apparatus comprising:

means for transmitting a plurality of service calls generated by a plurality of threads executed at the server to the client;

means for generating a synchronization call at the server, said synchronization call indicating that one of said plurality of threads executed at the server has changed and indicating a number of service calls generated by said plurality of threads at the server prior to the thread change;

means for transmitting said synchronization call to the client to allow the client to synchronize a service call execution;

means for receiving said synchronization call at the client; and

means for placing at least one of said service calls associated with said synchronization call into a wait position, if said number indicated in said synchronization call and said number of service calls executed at the client prior to receiving said synchronization call differ.